

11/03/00
JC914 U.S. PTO

PATENT
Docket No. 3553-4074US3

Express Mail Label No. EK599431149US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
UTILITY APPLICATION AND APPLICATION FEE TRANSMITTAL
(1.53(b))

914 U.S. PTO
09/708109
11/03/00

ASSISTANT COMMISSIONER FOR PATENTS

11-03-2000

U.S. Patent & TMO/TM Mail RptDt. #70

Sir:

Transmitted herewith for filing is the patent application of

Named Inventor(s): SCOTT NEDDERMAN

For: APPARATUS, SYSTEM, AND METHOD FOR VALIDATING NETWORK
COMMUNICATIONS DATA

Enclosed are:

☒ 24 page(s) of specification, 1 page(s) of Abstract, 4 pages of Claims 1-104

☒ 9 sheets of drawings [] formal [x] informal

[] 7 page(s) of Declaration and Power of Attorney

[] unsigned

[x] Newly Executed

[] Copy from prior application

[] Deletion of inventors including Signed Statement under 37 C.F.R. 1.63(d)(2)

[] Incorporation by Reference: The entire disclosure of the prior application, from which a copy of the combined declaration and power of attorney is supplied herein, is considered as being part of the disclosure of the accompanying application and is incorporated herein by reference.

[] Microfiche Computer Program (Appendix)

[] _____page(s) of Sequence Listing

[] computer readable disk containing Sequence Listing

[] Statement under 37 C.F.R. 1.821(f) that computer and paper copies of the Sequence Listing are the same

[] Claim for Priority

[] Certified Copy of the Priority Document

- ☐ English translation documents
- ☐ Information Disclosure Statement
- ☐ Copy of cited references
- ☐ Copy of PTO-1449
- ☐ Preliminary Amendment
- ☒ Return receipt postcard (MPEP 503)
- ☒ Assignment Papers (assignment cover sheet and assignment documents)
- ☐ A check in the amount of \$40.00 for recording the Assignment
- ☐ Assignment papers filed in parent application Serial No. _____
- ☐ Certification of chain of title pursuant to 37 C.F.R. 3.73(b)
- ☐ This is a ☐ continuation ☐ divisional ☐ continuation-in-part (C-I-P) of prior application serial no. _____.
- ☐ Cancel in the application original claims _____ of the parent application before calculating the filing fee. (At least one original independent claim must be retained for filing purposes).
- ☐ A Preliminary Amendment is enclosed. (Claims added by this Amendment have been properly numbered consecutively beginning with the number following the highest numbered original claim in the prior application).
- ☐ The status of the parent application is as follows:
- ☐ A Petition for Extension of Time and a Fee therefor has been and is being filed in the parent application to extend the term for action in the parent application until _____.
- ☐ A copy of the Petition for Extension of Time in the co-pending parent application is attached.
- ☐ No Petition for Extension of Time and Fee therefor are necessary in the co-pending parent application.
- ☐ Please abandon the parent application at a time while the parent application is pending or at a time when the petition for extension of time in that application is granted and while this application is pending has been granted a filing date, so as to make this application co-pending.
- ☐ Transfer the drawing(s) from the parent application to this application.
- ☐ Amend the specification by inserting before the first line the sentence:
This is a ☐ continuation ☐ divisional ☐ continuation-in-part of co-pending application Serial No. _____ filed _____.

I. CALCULATION OF APPLICATION FEE (For Other Than a Small Entity)

				Basic Fee
Number Filed		Number Extra	Rate	\$ 690.00
Total Claims	92	-20=	72	x\$18.00
Independent Claims	16	- 3=	13	x\$78.00
Multiple Dependent Claims				\$1,296.00
<input type="checkbox"/> yes				Additional Fee = \$260.00
<input checked="" type="checkbox"/> no				Additional Fee = NONE
				\$ -0-
TOTAL				\$3,000.00

☐ A statement claiming small entity status is attached or has been filed in the above-identified parent application and its benefits under 37 C.F.R. 1.28(a) is hereby claimed. Reduced fees under 37 C.F.R. 1.95(F) (50% of total) paid herewith \$_____.

☐ A check in the amount of _____ payment of the application fees is attached.


☒ Charge Fee(s) to Deposit Account No. 50-0712. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

☒ The Assistant Commissioner is hereby authorized to charge any additional fees which may be required for filing this application, or credit any overpayment to Deposit Account No. 50-0712. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

Respectfully submitted,

priceline.com Incorporated

Dated: November 3, 2000

By: 
John C. Andres
Registration No. 30,931

CORRESPONDENCE ADDRESS:

Mr. John C. Andres
priceline.com Incorporated
800 Connecticut Avenue
Norwalk, CT 06854

A

11/03/00
JCS75 U.S. PTO

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s) : SCOTT NEDDERMAN.

Serial No : TBA

Examiner: TBA

Filed :

For : APPARATUS, SYSTEM AND METHOD FOR VALIDATING NETWORK
COMMUNICATIONS DATA

09/708109
JCS75 U.S. PTO
11/03/00

EXPRESS MAIL CERTIFICATE



11-03-2000

U.S. Patent & TMO/TM Mail Rcpt Dt. #70

Express Mail Label No. EK599431149US

Date of Deposit NOVEMBER 3, 2000

I hereby certify that the following attached paper(s) and/or fee Utility Application and Fee Transmittal
(enclosing 24 pages of specification, 1 page of abstract, 14 pages of claims and 7 sheets of ~~formal~~ ^{Informal} drawings (Figs 1-
7), Declaration/Power of Attorney, Assignment, Assignment Transmittal Letter, and; Return Postcard is being
deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R.
§1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, United States Patent
and Trademark Office, 2900 Crystal Drive, Arlington, VA 22202-3513, ATTENTION: BOX PATENT
APPLICATIONS.

Lori E. Rooney

(Typed or printed name of person
mailing paper(s) and/or fee)

Lori E. Rooney

(Signature of person mailing
paper(s) and/or fee)

CORRESPONDENCE ADDRESS:

John C. Andres, Esq.
priceline.com Incorporated
800 Connecticut Avenue
Norwalk Connecticut 06854

Telephone: (203) 299-8671

RECEIVED
NOV - 9 2000
OPE/JCWS

543099_1

.94 5/15/00 12:27:52 PM

APPARATUS, SYSTEM, AND METHOD FOR VALIDATING NETWORK COMMUNICATIONS DATA

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims priority to U.S. provisional applications
5 60/177,463, filed January 20, 2000; 60/178,987, filed January 28, 2000; and 60/180,072,
filed February 3, 2000, each of which are hereby incorporated by reference.

FIELD

The present application relates generally to computer systems and software,
and more particularly to a method and system for the validation of information on a
10 communications network.

BACKGROUND

Typically, users (i.e. people or other systems) engage computers to facilitate
information processing. A computer operating system enables and facilitates users to access
and operate computer information technology. Somewhat like how automobile operator
15 interfaces (e.g. steering wheels and gearshifts) facilitate the access and operation of
automobile resources and functionality, computer user interfaces facilitate (e.g. windows and
cursors) the access and operation of computer hardware and operating system resources and
functionality. Graphical user interfaces such as the Apple Macintosh Operating System or
Microsoft's Windows 98 provide a baseline and means of accessing and displaying
20 information. Such consumer oriented operating systems enable users to access and operate
computer information technology by providing an integrated user interface. Other operating

systems such as Unix do not provide integrated graphical user interfaces and instead allow various interfaces to be employed such as command line interfaces (e.g. C-shell) and graphical user interfaces (e.g. X windows).

The proliferation and expansion of computer systems, databases, the Internet, and particularly the World Wide Web (the web), have resulted in a vast and diverse collection of information. Various user interfaces that facilitate the interaction of people with computer systems are currently in use. Tim Berners-Lee originally developed an information navigation interface called WorldWideWeb.app, i.e. the web, in late 1990 on NeXT Computer Inc.'s operating system, NeXTSTEP, at the European Organization for Nuclear Research (CERN, a particle physics center). Subsequently, information navigation interfaces, i.e. web browsers, have become widely available on almost every computer operating system platform.

Generally, the web is the manifestation and result of a synergetic interoperation between user interfaces (e.g. web browsers), servers, distributed information, protocols, and specifications. Web browsers were designed to facilitate navigation and access to information, while information servers were designed to facilitate provision of information. Typically, web browsers and information servers are disposed in communication with one another through a communications network; i.e. information servers typically provide information to users employing web browsers for navigating and accessing information about the web. Microsoft's Internet Explorer and Netscape Navigator are examples of web browsers. In addition, navigation user interface devices such as WebTV

have also been implemented to facilitate web navigation. Microsoft's Information Server and Apache are examples of information servers.

Information on the web typically is provided through and distributed employing a HyperText Markup Language (HTML) specification. HTML documents are also commonly referred to as web pages. HTML documents may contain links to other HTML documents that can be traversed by users of web browsers (i.e. user interfaces) by selecting the links, which are commonly highlighted by color and underlining. HTML has been extended and upgraded resulting in new standards such as Extensible Markup Language (XML) and other such variants, which provide greater functionality. HTML's progenitors were Standardized General Markup Language (SGML), which in turn was preceded by the General Markup Language (GML). SGML is generally regarded as a more functional superset of HTML and first appeared in 1980 as a draft by the Graphic Communications Association (GCA) to the American National Standards Institute (ANSI) (GCA 101-1983); it was adopted as an international standard by the International Standards Organization (ISO) in 1986 (ISO 8879:1986). Charles Goldfarb, Edward Mosher, and Raymond Lorie invented the GML at IBM to facilitate law office information system integration and improve document processing. GML itself was inspired by William Tunnicliffe, chairman of the CGA, during a presentation on the topic of "the separation of the information content of documents from their format" at the Canadian Printing Office in September, 1967.

HTML documents typically are accessed through navigation devices via a HyperText Transfer Protocol (HTTP). HTTP is a stateless application-level protocol for distributed, collaborative, hypermedia information systems, and is further described on the

W3C web site entitled HTTP Specifications and Drafts (available at

www.w3.org/Protocols/Specs.html). Microsoft's Information Server allows the tracking of a state with a built-in session object.

The basic web browsing paradigm presents users with a scrolling page full of
5 text, pictures, and various other forms of information media such as movies and links to other documents. Web browsers allow users to access uniquely identified HTML documents on the web by entering a navigation location in a Universal Resource Locator (URL) and employing HTTP as a transfer protocol to provide and obtain web pages. Typically, a user provides the address of a desired HTML document into a URL (either directly or through the
10 selection of links in an already viewed HTML document).

The onset of the web has brought on a tremendous increase in transactions occurring through insecure communications networks such as the Internet. Sending information through such networks allows for the possibility of data corruption. Database systems and other transactional systems benefit from the validation of data. Increasingly,
15 such transactions occur via web page forms. In many instances, it is critical that data acquired over a communications network be acquired in a consistent, predictable, and reliable way. Otherwise, a host of problems may result such as failed transactions, application errors, incorrect orders, irritated customers, and other such consequences.

Validation is a way to increase the reliability of data sent over a
20 communications network. Existing validation techniques are commonly employed to ensure the integrity of data against incidents such as transmission disruptions or erroneous data entry. Existing validation logic includes, for example, pattern matching logic to ensure

Attorney Docket No.: 3553-4074US3

Express Mail Label No.: XXXXXXXXXXXXUS

obtained data is in a proper form, and check sum techniques to ensure against data

corruption.

The most common way to perform validations on a form has been to write a multitude of small Javascript 1.0 functions that are downloaded to the client and then called

5 right before a form is submitted. The validating logic was imbedded into the web page itself.

A more recent and generic technique was developed to validate data from a form by describing the validations for each field and its dependencies and then calling generic validation procedures. The validation procedures are provided in an included file.

SUMMARY

As set forth below, a need exists for an improved apparatus, system, and method for improved validation of information across a communications network. Existing systems have the disadvantage of putting validation logic in a script into a web page resulting in an excess amount of validation logic that had to be downloaded with each page. Also, existing systems require the custom handling and development of validation logic on a per web page basis, thus greatly burdening and hindering development. Attempts to improve existing systems have failed to reduce the amount of validation logic downloaded by a user client, forcing them to download sometimes larger validation procedures in included files. Furthermore, the calling of validation procedures from an include file (i.e. a file, in this case containing large collections of validation logic, for wholesale inclusion) has not significantly improved development as the increasing number of exceptions require custom handling outside of common validation procedures.

The present application advances the art by providing a common validation rules library that may be accessed and executed either by a client or server. The present application advances the art by providing a minimal amount of logic either to the client or the server by generating only a requisite amount of validation logic. Furthermore, the present application improves development by allowing developers to reuse and build upon validation logic.

In one embodiment, the present system includes an apparatus, comprising: a memory device having at least one region for storing executable program code; and a processor, disposed in communication with the memory device, for executing the program

code stored in the memory device. The program code further comprises code to receive

information over a communications network, code to retrieve validation rules from a rules

library, and code to determine data validity by applying the retrieved validation rules to the

information.

5 In another embodiment, the present system involves an apparatus, comprising:

a memory device having at least one region for storing executable program code; and a

processor, disposed in communication with the memory device, for executing the program

code stored in the memory device. The program code further comprises code to provide

validation rules from a rules library, and code to validate information with provided

10 validation rules.

In other embodiments, the present system involves (1) code to provide a rules

library, and code to build validation rules by subclassing members of a rules library class

hierarchy, (2) code to mark data types for associated validation rules, and code to provide

validation marked data types, and/or (3) code to identify browser capability, code to choose

15 improved validation deployment, code to provide appropriate network location and

validation rules, code to obtain information from user, and code to validate information with

appropriate validation rules.

20

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings illustrate certain embodiments of the application.

Figure 1 illustrates a centralized controller according to one embodiment of the present application;

5 Figure 2 illustrates another embodiment of the present application in the form of a distributed system interacting through a communications network;

Figure 3 illustrates another embodiment of the system and various interactions;

Figure 4 illustrates web pages, hypertext, reference and proximal links;

10 Figure 5 is a flowchart illustrating a validator refinement system;

Figure 6 is a flowchart illustrating an improved validation chooser system;

Figure 7 is a flowchart illustrating an improved client validator;

Figure 8 is a flowchart illustrating an improved server validator;

15 Figure 9 is a flowchart illustrating one embodiment of a validation rules inheritance hierarchy.

DETAILED DESCRIPTION

Figure 1 shows one embodiment of a system incorporating the present application. In this embodiment, the system includes a centralized controller 1101 configured to receive information from one or more users from user input device(s) 1114.

5 Also, the centralized controller may receive information from a communications network 1115 through its input/output (I/O) facility 1105, preferably, via a network interface 1107. The I/O facility is capable of both receiving and sending information. Peripheral devices 1116 may be attached to the centralized controller for any number of purposes including, but not limited to: printers for output, scanners for input, additional or alternative storage devices
10 for data storage and retrieval, network interfaces for communication, and devices of the like.

A typical centralized controller may be based on common computer systems that may include, but are not limited to, components such as: a central processing unit (CPU) 1104, random access memory (RAM) 1103, read only memory 1102, and a local storage device 1108. The CPU is electronically coupled to each of the central controller's other
15 elements. The CPU comprises at least one high-speed data processor adequate to execute program modules for executing user or system-generated requests. These modules are described in Figures 2 through 9. Preferably, the CPU is a conventional microprocessor such as the Intel Pentium Processor. The CPU interacts with RAM, ROM, and storage device(s) to execute stored program code according to conventional data processing techniques.

20 The local storage device may contain modules. These modules may include, but are not limited to, a network communications data validator (NCDV) 1109, a user interface 1110, an operating system 1111, a web browser 1112, an information server 1106,

and a database 1113. These modules may be stored and accessed from the local storage

device(s) or from storage devices accessible through I/O. Although these modules typically and preferably are stored in a local storage device, they may also be stored in ROM, RAM, peripheral devices or in remote storage facilities through a communications network.

5 The operating system is executable program code enabling the operation of a centralized controller. The operating system facilitates access of storage devices, I/O, network interfaces devices, peripheral devices, etc. The operating system preferably is a conventional product such as a Unix operating system or Microsoft Windows NT. The operating system, once executed by the CPU, interacts with ROM, RAM, I/O, peripheral
10 devices, user input devices, storage devices, communications networks, program modules, and data, et al. Preferably, the operating system includes communication protocols that allow the centralized controller to communicate with other entities through a communications network. The preferable protocol is TCP/IP.

 Figure 2 shows another embodiment of a system incorporating the present
15 application. In this embodiment, the centralized controller 1101 embodiment of Figure 1 has been decentralized into components: a user interface controller 2201 or alternatively a user interface device 2202, a network communications data validator controller 2203, a web browser controller 2204, a database controller 2205, and an information server controller 2206.

20 A user interface controller is configured similarly to the centralized controller of Figure 1 except it does not require a database, NCDV, information server, or web browser. A user interface 2110 is stored program code that is executed by the CPU. The user interface

is responsible for receiving either user or system-generated requests. The user interface

device may be a telephone, a consumer electronics online access device (e.g. Phillips Inc.'s WebTV), PDA or the like. In alternative embodiments, a user interface device 2202 may take the place of or be used in conjunction with a user interface controller.

5 In one embodiment, a centralized server 1101 is configured as a web server, and conventional communications software such as Netscape Navigator web browser may be used to transmit a conditional purchase offer (CPO). In one embodiment, the CPO centralized server 1101 has a web page on the world wide web, allowing the buyer to provide information through the interface of the conventional web browser software. In one
10 embodiment, the buyer selects the subject of the goods he wishes to purchase by selecting from a list of possible subjects. Subjects might include airline tickets, hotel rooms, rental cars, insurance, mortgages, clothing, etc. After the subject is selected, a form is displayed on a video monitor of a buyer interface. This form is an electronic contract with a number of blanks to be filled out by the buyer, with each blank representing a condition of a CPO.

15 An information server controller is configured similarly to the centralized controller of Figure 1 except it does not require a NCDV, database, web browser, or user interface. An information server 2106 is a software server that is stored program code that is executed by the CPU. Preferably, the information server is a conventional Internet information server such as Microsoft's Internet Information Server version 4.0. Preferably,
20 the information server allows for the execution of program modules through facilities such as C++, Java, JavaScript, ActiveX, CGI scripts, ASP, or any like facility with regular expression (regex) abilities on the server side. An information server typically takes requests

from a web browser and provides results to a web browser; however, an information server

can take requests from user interfaces as well. The information server may also take system requests. In alternative embodiments, a information server may be integrated into a user interface or vice versa, thus, combining the functionality of both.

5 A network communications data validator controller is configured similarly to the centralized controller of Figure 1 except it does not require a database, web browser, information server, or user interface. The NCDV 2109 is stored program code that is executed by the CPU. A NCDV takes requests from a user interface and provides results to a user interface. The NCDV may also take system requests. In Figure 1 the NCDV is
10 implemented within an information server, however, the NCDV may also be implemented independently and interact with an information server through provided APIs as illustrated in Figure 2 2106 and 2109.

 A database controller is configured similarly to the centralized controller of Figure 1 except it does not require a NCDV, web browser, information server, or user
15 interface. A database(s) 2113 is stored program code that is executed by the CPU and it is stored data processed by the CPU. A database takes requests from a NCDV and provides results to a NCDV. The database may also take system requests. In an alternative embodiment, a NCDV may be integrated into a database or vice versa, thus, combining the functionality of both. In yet another alternative embodiment, a NCDV may be integrated into
20 a user interface or vice versa, thus, combining the functionality of both.

 A web browser controller is configured similarly to the centralized controller of Figure 1 except it does not require a NCDV, database, information server, or user

interface. A web browser 2112 is stored program code that is executed by the CPU.

Preferably, the web browser is a conventional hypertext viewing application such as Microsoft Internet Explorer or Netscape Navigator. Preferably, the web browser allows for the execution of program modules through facilities such as Java, JavaScript (preferably revision number 1.2 or greater), ActiveX or any like facility with regular expression (regex) abilities. A web browser takes requests from a user interface and provides results to a user interface. The web browser may also take system requests. In alternative embodiments, a web browser may be integrated into a user interface or vice versa, thus, combining the functionality of both.

The functionality of the user interface controller, network communications data validator controller, database controller, information server controller, and web browser controller may be combined in any number of ways to facilitate deployment. To accomplish this, one may simply copy the executable code, first ensuring it has been compiled for the appropriate CPU of the controller for which it is destined, and/or data onto local storage device of one of the various controllers. Similarly, the functionality of the user interface, NCDV, database, information server, and web browser may be combined in any number of ways to facilitate deployment. To accomplish this, one must simply integrate the components into one code base or in a facility that can dynamically load the components on demand in an integrated fashion.

Figure 3 shows an overview of the basic interaction of the system. The information server 3106 acts as an in-between for: a user interface 3110 on a system, a user interface device 3202, or a web browser 3112 taking requests. The information server can

make further requests of a NCDV 3109 that itself may access one or more databases or files

such as a rules library. Both the information server and the NCDV may service multiple

instances of any of the aforementioned. Also, there may be one or more instances of the

NCDV and/or information server that may severally or jointly interact with one or more

5 information servers 3106. NCDVs service information servers. In turn, the information

servers and/or NCDVs may interact and service one or more databases 3113 for various

purposes such as, but not limited to: validation rules, session state storage, and electronic

commerce (E-commerce) order fulfillment. Figure 3 shows that the NCDV and information

server may service multiple sources at once, and that the NCDV and information server may

10 access more than one database.

Figure 4 shows web pages 4401 with hypertext 4402 and reference links 4403

at various navigation locations 4404. An originating navigation location 4404a references

hypertext that may have initial reference links 4403a. These initial reference links are

proximal links to the originating navigation location.

15 One may view hypertext at an initial reference navigation location 4404b by

traversing an initial reference link. The subsequent reference links 4403b found in the

hypertext found at the initial reference navigation location are also proximal links, however,

they are one reference less proximal (i.e. one "hop" away) to the originating navigation

location.

20 One may view hypertext at a subsequent reference navigation location 4404c

by traversing a subsequent reference link. The further subsequent reference links 4403c

found in the hypertext found at the subsequent reference navigation location are also

proximal links, however, they are two references less proximal (i.e. two “hops” away) to the originating navigation location.

Figure 5 outlines a validator refinement system. Initially, a user, or even an automated system such as a web bot, navigates a communications network 5501, for example the Internet. Navigation typically occurs via a user interface, user interface device, or web browser. A user typically navigates by positioning a pointing element 4405 of Figure 4 by means of a positioning device such as, but not limited to, a mouse. Typically this is referred to as “surfing the Internet,” “surfing the net,” or simply “surfing.”

Web browsers and the like keep track of the navigation location they are visiting, and can provide this information to other facilities through various application program interfaces (API)s. Web browsers’ may provide their name, revision, enabled and disabled features, similarly, through APIs; also, it is possible to determine relevant feature sets of almost any web browser from its name and revision number.

Upon surfing to a navigation location with the ability to identify a web browser, the NCDV or the information server can thus identify web browser capabilities 5502 through a provided API; for example, Javascript provides methods such as navigator.javaEnabled, navigator.appName, and navigator.appVersion for identifying the abilities, type and revision number of an employed web browser. Thus, a web page can be constructed to obtain such information, and may then further provide the NCDV or information server with the web browser identity.

Upon having identified the web browser, overall performance may be improved by choosing an appropriate validation scheme 5503. Various web browsers have

equally varying capabilities that affect performance. By identifying web browser

capabilities, the validator refinement system may determine which, among available, validation schemes are best.

Upon having determined which validation scheme is best for the identified
5 web browser, the validator refinement system provides the user with a navigation location at which web pages are tailored with the most appropriate validation schemes 5504.

After the user navigates to the appropriate web page with appropriate validation scheme, the user may provide information into the web page 5505. In one
embodiment, such provision of information is facilitated by employing web forms, however,
10 other standard data processing techniques may also be used.

Upon provision of information by the user, the appropriate validation scheme will validate the information 5506.

Figure 6 outlines an improved validator chooser system, further explaining how an improved (i.e. regex enabled) validation scheme is chosen 5503 of Figure 5. Upon
15 having identified the browser capability 5502 of Figure 5, the improved validator chooser system will determine, using standard data processing techniques, if the browser's supported languages, if any, support regex 6601. In one embodiment, this may be accomplished by embedding commands into a web page that will be executed upon users' traversal if their browser supports the processing of such commands. Any number of command types may be
20 used including, but not limited to: Javascript, Active X, Java, and others of the like. In one embodiment, Javascript will be employed.

If users' browsers support regex, then their information provisions will be validated with an improved validator scheme on the client 6602; typically the client may take the form of a user interface such as, but not limited to a web browser (e.g. Netscape Navigator), or a user interface device (e.g. Phillips WebTV). Thus, the user will be provided with a network location 5504 of Figure 5 that employs an improved client validator scheme. However, if users' browsers do not support regex, and there is no other (non regex) language facility provided by the web browser 6603, then users' information will be validated with an improved validator scheme on the server 6605. Thus, the user will be provided with a network location 5504 of Figure 5 that employs an improved server validator scheme. If, however, users' browsers do not support regex, and there is a (non regex) language facility provided by the web browser 6603, then users' information will be validated with traditional validation techniques 6604; traditional validation techniques such as monolithic include files and non regex based validation via languages such as ActiveX. Thus, the user will be provided with a network location 5504 of Figure 5 that employs traditional validation schemes.

Figure 7 outlines an improved client validator system, further explaining how an improved validation scheme is applied to user information on a client when a browser supports regex 6602 of Figure 6. Based on the abilities of users' web browsers, a web page is requested 7701. The information server provides and/or generates the requested page 7702 via standard web server techniques.

Upon the improved client validator system's provision of the requested web page, the information server provides and/or generates validation code from a rules library.

This validation code is imbedded into the requested web page. Preferably this validation

code is Javascript revision 1.2 or greater, however, this code may be provided through other like facilities such as, but not limited to: Microsoft Visual Basic Script, Java applets, and others of the like. The information server provides rules from the rules library associated

5 with web page field types; e.g. if the web page is marked as an airTravel type of page, a corresponding airTravel set of rules will be pulled from a rules library and imbedded into the requested web page 7703. Those skilled in the art will recognize that the “airTravel” notation denotes an object in objected-oriented development parlance; specifically objects are demarcated by concatenating multiple words without spaces, and capitalizing all
10 concatenated words except for the first word. Ergo, “airTravel” might be an example of an object concerned with the validation of air travel data entry.

Upon provision of the requested web page with the appropriate imbedded rules, the resulting web page is provided to the user 7704. Unlike with traditional techniques such as monolithic include files, only the validation logic required for the resulting web page
15 is provided. Typically this results in the user viewing the resulting page in their web browser with the appropriate rules activated.

Upon provision of the resulting web page to the user, the user may provide information specific to the web page 7705. This information may be of any nature.

Examples include, but are not limited to, order entry of information required to complete an
20 online purchase of goods or services. Typically, the user simply enters information into provided fields.

As the user provides information, imbedded validation code validates the entries 7706. This validation may occur as the entry is being provided, or after a set of entries has been provided.

If the provided user information is not valid 7707, then the current web page
5 will identify invalid entries with an error message 7708. In one embodiment this is done by highlighting the invalid entries employing standard HTML modification techniques. After the invalid information is identified, the user may edit their entries 7705.

Upon the user's provision of valid information 7707, the improved client
validator will check if there are anymore entries required to complete an entry transaction
10 7709. If there are more entries required, the user is provided with the next form web page 7704. In one embodiment, all the rules and all the required web form sections are provided to the user in a single web page provision upon the first web page provision to the user 7704. Thus, subsequent cycling from 7707 to 7709 of multi-part web page forms require no interaction from an information server. Upon reaching the end of a web form 7709, the
15 validated user information is posted to the information server 7710. In an alternative embodiment, validated user information is posted to the information server before the entry completion check 7709 and after the user's provision of valid information 7707.

Figure 8 outlines an improved server validator system, further explaining how an improved validation scheme is applied to user information on a server when a browser
20 does not support regex 6605 of Figure 6. Based on the abilities of a user's web browsers, a web page is requested 8801. Of course, as described in Figure 3 single and/or multiple users may make multiple uses of a single or multiple server validator systems. The information

server provides and/or generates the requested page 8802 via standard web server techniques.

Typically this results in the user viewing the resulting page in their web browser.

Upon the improved server validator system's provision of the requested web page, the user may provide information specific to the web page, thereafter posting their

5 information to the information server 8803. This information may be of any nature. An Example includes, but is not limited to, the order entry of information required to complete an online purchase of goods or services. Typically, the user simply enters information into provided fields.

Upon having posted the user information to the information server, the
10 improved server validator system validates the user information 8804 by accessing a rules library made available to the server. The rules library makes validation code available to the improved server validator. Preferably this validation code is Javascript revision 1.2 or greater, however, this code may be provided through other like facilities such as, but not limited to: Microsoft Visual Basic Script, Java applets, C++, and others of the like.

15 Employing Javascript revision 1.2 or other such high level object oriented languages is believed to be preferable for savings in development time because a single development may be employed by both clients and servers, and furthermore inheritance techniques may be employed with greater facility. The information server provides rules from the rules library associated with web page field types; e.g. if the web page is marked as an airTravel type of
20 page, a corresponding airTravel set of rules will be pulled from a rules library. However, unlike the improved client validator, no rules are imbedded into the web page. Instead, the rule code is executed on the server(s)' CPU(s).

If the provided user information is not valid 8805, then the current web page will identify invalid entries with an error message 8806. In one embodiment this is done by highlighting the invalid entries employing standard HTML modification techniques. Upon having modified the web page highlighting the errors, the information server provides the
5 modified web page to the user 8802 and the user may edit their entries 8803.

Upon the user's provision of valid information 8805, the improved server validator will check if there are anymore entries required to complete an entry transaction 8807. If there are more entries required, the information server provides the user with the next form web page 8802. Thus, subsequent cycling from 8802 to 8807 of multi-part web
10 pages are provided by the information server. Upon reaching the end of a web form 8807, the validated user information is posted to the information server 8808. In an alternative embodiment, validated user information is posted to the information server before the entry completion check 8807 and after the user's provision of valid information 8805.

Figure 9 outlines a rules library system (i.e. rules library), further explaining
15 how the improved validators (Figure 7 and Figure 8) perform validation. The rules library may be built with conventional development design tools; in one embodiment object oriented tools such as, but not limited to: Java, Javascript, C++, Objective C, and others of the like. However, one skilled in the art will recognize that any turing complete language and/or development environment may be alternatively employed. Preferably a language such as
20 Javascript revision 1.2 is employed as it may run both on clients and on the server without modification to the rules library allowing for greater code reuse. In the preferred embodiment, the rules library is a central repository of validation rules organized in a class

hierarchy as shown in Figure 9. However, the rules library in alternative embodiments may be organized as collection(s) of files, entries in a database, and/or other structures of the like.

The class hierarchy is born from a parent 9901. A parent class may have children 9902.

Children are born, i.e. subclassed, from their parent. In turn, the children may also be

5 subclassed and have their own children 9903, 9904. Subclassing provides children with an inheritance. The parent may have certain attributes 9906a in the form of instance variables (or other such standard data processing facilities) and abilities 9907a in the form of methods (or other such standard data processing facilities). When a parent is subclassed, its child inherits all of its attributes and abilities. Furthermore, the child may further redefine itself
10 from its parent with additional attributes 9906b and abilities 9907b. This class hierarchy lineage may be reflected by illustrating the inheritance source 9905.

In one alternative embodiment, the validation rules may be associated with data types by marking data types with associated validation rules. Creating an association between data types (e.g. XML, web form fields, and or the like) and validation rules facilitate
15 the application of validation rules upon a desired set of data. Creating the association reduces stress on computing resources by not having to first parse a web page to determine what types of validation rules ought to be applied. The creation of an association between data types and validation rules may be accomplished using standard data processing techniques such as, but not limited to: tagging web form elements by employing extensible
20 markup language (XML) sheets. Associations with data types may be created by marking data types such as, but not limited to, fields with tags that may be used to locate and/or provide appropriate validation logic, i.e. an appropriate validation rule, from a rules library.

In another alternative embodiment, a parser may examine the structure and provide

appropriate validation logic based on the parser's best guess of the data type.

The provision of a rules library provides a single source for the improved client validator and the improved server validator for accessing validation rules. Although
5 there may be a single rules library source, employing standard information technology, the rules library may be maintained in multiple locations for redundancy and performance purposes. A single source allows for the reuse of validation rules and thereby reduces development time. Also, allowing children to inherit from parent rules further reduces the time for developing new rules. For example, in Figure 9, travelRules 9902c inherits the
10 ability to validate and encrypt 9907a various forms of customer information 9906a. Thus, when defining the new child, travelRules, there is no need to provide such attributes and abilities as they are automatically provided through inheritance. However, the travelRules child may further define departure and destination attributes 9906b and the ability to transform 9907b attributes; e.g. the ability to transform valid user time entries from a twelve
15 hour time notation into a twenty-four hour notation or vice versa may be desirable. Furthermore, the child itself, travelRules, may be subclassed and further defined by airTravelRules adding more attributes 9906c such as class, windowSeat, etc.

This rules library may be accessed by the information server through standard APIs employing standard data processing techniques. In one embodiment, web page forms
20 are tagged with a type marker in HTML. These type markers will allow the information server to employ an associated rule.

Attorney Docket No.: 3553-4074US3

Express Mail Label No.: XXXXXXXXXXXXUS

It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this application and that various modifications may be implemented without departing from the scope and spirit of the application.

CLAIMS

What is claimed is:

- 1 1. A method, comprising:
2 receiving information over a communications network;
3 retrieving validation rules from a rules library stored in a memory device; and
4 determining computer data validity by applying the retrieved validation rules to the
5 information.
- 1 2. The method of claim 1, further comprising highlighting information determined to be
2 invalid by the validation rules.
- 1 3. The method of claim 1, wherein validation rules are provided to a client.
- 1 4. The method of claim 1, wherein validation rules are provided to a server.
- 1 5. The method of claim 1, wherein validation rules are imbedded into a web page.
- 1 6. The method of claim 1, wherein validation rules are executable both on a client and
2 server.
- 1 7. A system, comprising:
2 means for receiving information over a communications network;
3 means for retrieving validation rules from a rules library stored in a memory device;
4 and
5 means for determining computer data validity by applying the retrieved validation
6 rules to the information.
- 1 8. The system of claim 7, further comprising means for highlighting information
2 determined to be invalid by the validation rules.

- 1 9. The system of claim 7, wherein validation rules are provided to a client.
- 1 10. The system of claim 7, wherein validation rules are provided to a server.
- 1 11. The system of claim 7, wherein validation rules are imbedded into a web page.
- 1 12. The system of claim 7, wherein validation rules are executable both on a client and
2 server.
- 1 13. Computer executable software code stored on a computer readable medium, the code,
2 comprising:
3 code for receiving information over a communications network;
4 code for retrieving validation rules from a rules library stored in a memory device;
5 and
6 code for determining computer data validity by applying the retrieved validation rules
7 to the information.
- 1 14. The medium of claim 13, further comprising code for highlighting information
2 determined to be invalid by the validation rules.
- 1 15. The medium of claim 13, wherein validation rules are provided to a client.
- 1 16. The medium of claim 13, wherein validation rules are provided to a server.
- 1 17. The medium of claim 13, wherein validation rules are imbedded into a web page.
- 1 18. The medium of claim 13, wherein validation rules are executable both on a client and
2 server.

- 1 19. An apparatus, comprising:
- 2 a memory device having at least one region for storing executable program code; and
- 3 a processor, disposed in communication with the memory device, for executing the
- 4 program code stored in the memory device, wherein the program code, further comprising:
- 5 code to receive information over a communications network;
- 6 code to retrieve validation rules from a rules library stored in a memory
- 7 device;
- 8 code to determine computer data validity by applying the retrieved validation
- 9 rules to the information.
- 1 20. The apparatus of claim 19, further comprising code to highlight information
- 2 determined to be invalid by the validation rules.
- 1 21. The apparatus of claim 19, wherein validation rules are provided to a client.
- 1 22. The apparatus of claim 19, wherein validation rules are provided to a server.
- 1 23. The apparatus of claim 19, wherein validation rules are imbedded into a web page.
- 1 24. The apparatus of claim 19, wherein validation rules are executable both on a client
- 2 and server.
- 1 25. A method, comprising:
- 2 identifying data types requiring validation; and
- 3 providing validation rules stored in a memory device for the associated data types
- 4 from a rules library.
- 1 26. The method of claim 25, wherein validation rules are provided to a client.
- 1 27. The method of claim 25, wherein validation rules are provided to a server.

- 1 28. The method of claim 25, wherein validation rules are imbedded into a web page.
- 1 29. The method of claim 25, wherein validation rules are executable both on a client and
2 server.
- 1 30. A system, comprising:
2 means for identifying data types requiring validation; and
3 means for providing validation rules stored in a memory device for the associated
4 data types from a rules library.
- 1 31. The system of claim 30, wherein validation rules are provided to a client.
- 1 32. The system of claim 30, wherein validation rules are provided to a server.
- 1 33. The system of claim 30, wherein validation rules are imbedded into a web page.
- 1 34. The system of claim 30, wherein validation rules are executable both on a client and
2 server.
- 1 35. Computer executable software code stored on a computer readable medium, the code,
2 comprising:
3 code for identifying data types requiring validation; and
4 code for providing validation rules stored in a memory device for the associated data
5 types from a rules library.
- 1 36. The medium of claim 35, wherein validation rules are provided to a client.
- 1 37. The medium of claim 35, wherein validation rules are provided to a server.
- 1 38. The medium of claim 35, wherein validation rules are imbedded into a web page.
- 1 39. The medium of claim 35, wherein validation rules are executable both on a client and
2 server.

- 1 40. An apparatus, comprising:
- 2 a memory device having at least one region for storing executable program code; and
- 3 a processor, disposed in communication with the memory device, for executing the
- 4 program code stored in the memory device, wherein the program code, further comprising:
- 5 code to identify data types requiring validation;
- 6 code to provide validation rules stored in a memory device for the associated
- 7 data types from a rules library.
- 1 41. The apparatus of claim 40, wherein validation rules are provided to a client.
- 1 42. The apparatus of claim 40, wherein validation rules are provided to a server.
- 1 43. The apparatus of claim 40, wherein validation rules are imbedded into a web page.
- 1 44. The apparatus of claim 40, wherein validation rules are executable both on a client
- 2 and server.
- 1 45. A method, comprising:
- 2 providing a rules library and an initial parent rule stored in a memory device; and
- 3 building validation rules by subclassing members of a rules library class hierarchy.
- 1 46. The method of claim 45, further comprising storing subclassed validation rules in a
- 2 rules library.
- 1 47. The method of claim 45, wherein the subclassed validation rules inherit validation
- 2 logic from a parent rule.
- 1 48. The method of claim 45, wherein validation rules are associated with data types.
- 1 49. The method of claim 45, wherein validation rules are imbedded into a web page.

1 50. The method of claim 45, wherein validation rules are executable both on a client and
2 server.

1 51. A system, comprising:

2 means for providing a rules library and an initial parent rule stored in a memory
3 device; and

4 means for building validation rules by subclassing members of a rules library class
5 hierarchy.

1 52. The system of claim 51, further comprising means for storing subclassed validation
2 rules in a rules library.

1 53. The method of claim 51, wherein the subclassed validation rules inherit validation
2 logic from a parent rule.

1 54. The system of claim 51, wherein validation rules are associated with data types.

1 55. The system of claim 51, wherein validation rules are imbedded into a web page.

1 56. The system of claim 51, wherein validation rules are executable both on a client and
2 server.

1 57. Computer executable software code stored on a computer readable medium, the code,
2 comprising:

3 code for providing a rules library and an initial parent rule stored in a memory device;
4 and

5 code for building validation rules by subclassing members of a rules library class
6 hierarchy.

1 58. The medium of claim 57, further comprising code for storing subclassed validation
2 rules in a rules library.

1 59. The method of claim 57, wherein the subclassed validation rules inherit validation
2 logic from a parent rule.

1 60. The medium of claim 57, wherein validation rules are associated with data types.

1 61. The medium of claim 57, wherein validation rules are imbedded into a web page.

1 62. The medium of claim 57, wherein validation rules are executable both on a client and
2 server.

1 63. An apparatus, comprising:
2 a memory device having at least one region for storing executable program code; and
3 a processor, disposed in communication with the memory device, for executing the
4 program code stored in the memory device, wherein the program code, further comprising:
5 code to provide a rules library and an initial parent rule stored in a memory
6 device;
7 code to build validation rules by subclassing members of a rules library class
8 hierarchy.

1 64. The apparatus of claim 63, further comprising code to store subclassed validation
2 rules in a rules library.

1 65. The method of claim 63, wherein the subclassed validation rules inherit validation
2 logic from a parent rule.

1 66. The apparatus of claim 63, wherein validation rules are associated with data types.

1 67. The apparatus of claim 63, wherein validation rules are imbedded into a web page.

1 68. The apparatus of claim 63, wherein validation rules are executable both on a client
2 and server.

1 69. A method, comprising:
2 marking data types for associated validation rules from a rules library stored in a
3 memory device; and
4 providing validation marked data types.

1 70. The method of claim 69, further comprising building forms with validation rules
2 associated with marked data types.

1 71. The method of claim 69, further comprising storing forms with validation rules
2 associated with marked data types.

1 72. The method of claim 69, further comprising providing forms with validation rules
2 associated with marked data types over a communications network.

3 73. The method of claim 69, wherein validation rules are imbedded into a web page.

1 74. The method of claim 69, wherein validation rules are executable both on a client and
2 server.

1 75. A system, comprising:
2 means for marking data types for associated validation rules from a rules library
3 stored in a memory device; and
4 means for providing validation marked data types.

1 76. The system of claim 75, further comprising means for building forms with validation
2 rules associated with marked data types.

1 77. The system of claim 75, further comprising means for storing forms with validation
2 rules associated with marked data types.

1 78. The system of claim 76, further comprising means for providing forms with
2 validation rules associated with marked data types over a communications network.

3 79. The system of claim 75, wherein validation rules are imbedded into a web page.

1 80. The system of claim 75, wherein validation rules are executable both on a client and
2 server.

1 81. Computer executable software code stored on a computer readable medium, the code,
2 comprising:

3 code for marking data types for associated validation rules from a rules library stored
4 in a memory device; and

5 code for providing validation marked data types.

1 82. The medium of claim 81, further comprising code for building forms with validation
2 rules associated with marked data types.

1 83. The medium of claim 81, further comprising code for storing forms with validation
2 rules associated with marked data types.

1 84. The medium of claim 82, further comprising code for providing forms with validation
2 rules associated with marked data types over a communications network.

3 85. The medium of claim 81, wherein validation rules are imbedded into a web page.

1 86. The medium of claim 81, wherein validation rules are executable both on a client and
2 server.

- 1 87. An apparatus, comprising:
- 2 a memory device having at least one region for storing executable program code; and
- 3 a processor, disposed in communication with the memory device, for executing the
- 4 program code stored in the memory device, wherein the program code, further comprising:
- 5 code to mark data types for associated validation rules from a rules library
- 6 stored in a memory device;
- 7 code to provide validation marked data types.
- 1 88. The apparatus of claim 87, further comprising code to build forms with validation
- 2 rules associated with marked data types.
- 1 89. The apparatus of claim 87, further comprising code to store forms with validation
- 2 rules associated with marked data types.
- 1 90. The apparatus of claim 88, further comprising code to provide forms with validation
- 2 rules associated with marked data types over a communications network.
- 1 91. The apparatus of claim 87, wherein validation rules are imbedded into a web page.
- 1 92. The apparatus of claim 87, wherein validation rules are executable both on a client
- 2 and server.

- 1 93. A method, comprising:
- 2 identifying browser capability;
- 3 choosing a validation deployment, wherein the validation deployment comprising:
- 4 determining if a browser supports regular expressions, and if so, providing
- 5 validation rules to a client;
- 6 determining if the browser supports non regular expression language, and if
- 7 so, providing non regular expression language information validation;
- 8 determining if the browser does not support non regular expression language,
- 9 and if not, providing regex enabled validation on a server;
- 10 providing the browser with appropriate network location and validation rules;
- 11 obtaining information from a user; and
- 12 validating information with appropriate validation rules stored in a memory device.
- 1 94. The method of claim 93, wherein validation rules are imbedded into a web page.
- 1 95. The method of claim 93, wherein validation rules are executable both on a client and
- 2 server.

1 96. A system, comprising:

2 means for identifying browser capability;

3 means for choosing a validation deployment, wherein the validation deployment
4 comprising:

5 means for determining if a browser supports regular expressions, and if so,
6 providing validation rules to a client;

7 means for determining if the browser supports non regular expression
8 language, and if so, providing non regular expression language information validation;

9 means for determining if the browser does not support non regular expression
10 language, and if not, providing regex enabled validation on a server;

11 means for providing the browser with appropriate network location and validation
12 rules;

13 means for obtaining information from a user; and

14 means for validating information with appropriate validation rules stored in a memory
15 device.

1 97. The system of claim 96, wherein validation rules are imbedded into a web page.

1 98. The system of claim 96, wherein validation rules are executable both on a client and
2 server.

99. Computer executable software code stored on a computer readable medium, the code,

comprising:

code for identifying browser capability;

code for choosing a validation deployment, wherein the validation deployment

comprising:

code for determining if a browser supports regular expressions, and if so,

providing validation rules to a client;

code for determining if the browser supports non regular expression language,

and if so, providing non regular expression information validation;

code for determining if the browser does not support non regular expression

language, and if not, providing regex enabled validation on a server;

code for providing the browser with appropriate network location and validation

rules;

code for obtaining information from a user; and

code for validating information with appropriate validation rules stored in a memory

device.

100. The medium of claim 99, wherein validation rules are imbedded into a web page.

101. The medium of claim 99, wherein validation rules are executable both on a client and server.

1 102. An apparatus, the code, comprising:

2 a memory device having at least one region for storing executable program code; and

3 a processor, disposed in communication with the memory device, for executing the

4 program code stored in the memory device, wherein the program code, further comprising:

5 code to identify browser capability;

6 code to choose a validation deployment, wherein the validation deployment

7 comprising:

8 code to determine if a browser supports regular expressions, and if so,

9 provide validation rules to a client;

10 code to determine if the browser supports non regular expression

11 language, and if so, provide non regular expression information validation;

12 code to determine if the browser does not support non regular

13 expression language, and if not, provide regex enabled validation on a server;

14 code to provide the browser with appropriate network location and validation

15 rules;

16 code to obtain information from a user; and

17 code to validate information with appropriate validation rules stored in a

18 memory device.

1 103. The apparatus of claim 102, wherein validation rules are imbedded into a web page.

1 104. The apparatus of claim 102, wherein validation rules are executable both on a client

2 and server.

ABSTRACT

An apparatus, system, and method for the improved validation of information across a communications network provide a common validation rules library that may be accessed and executed either by a client or server. The provision of a minimal amount of logic either to the client or the server is achieved by generating only a requisite amount of validation logic. Furthermore, developers are enabled to reuse and build upon validation logic by subsequent validation logic inheriting validation logic from a parent rule.

Figure 1

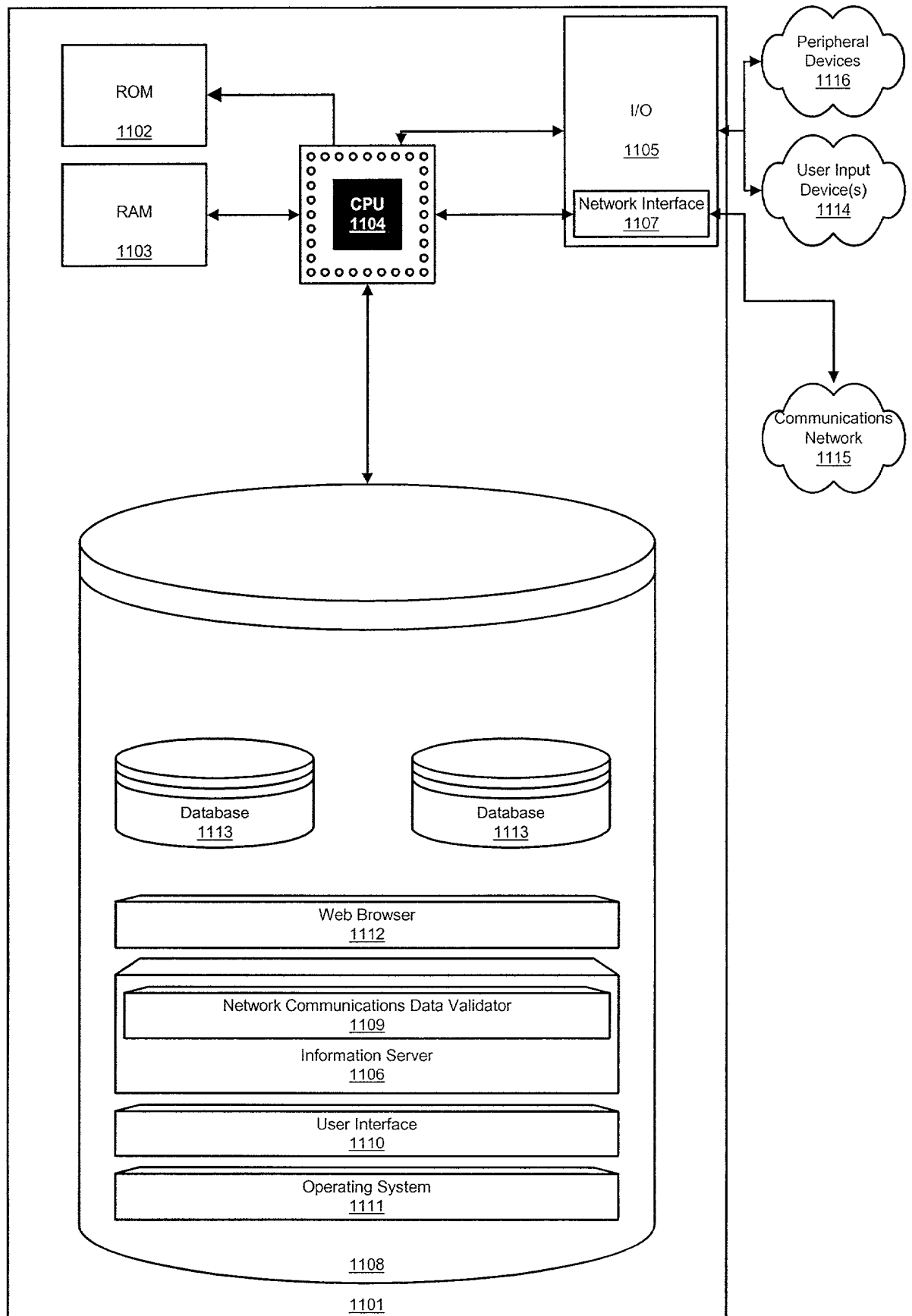


Figure 2

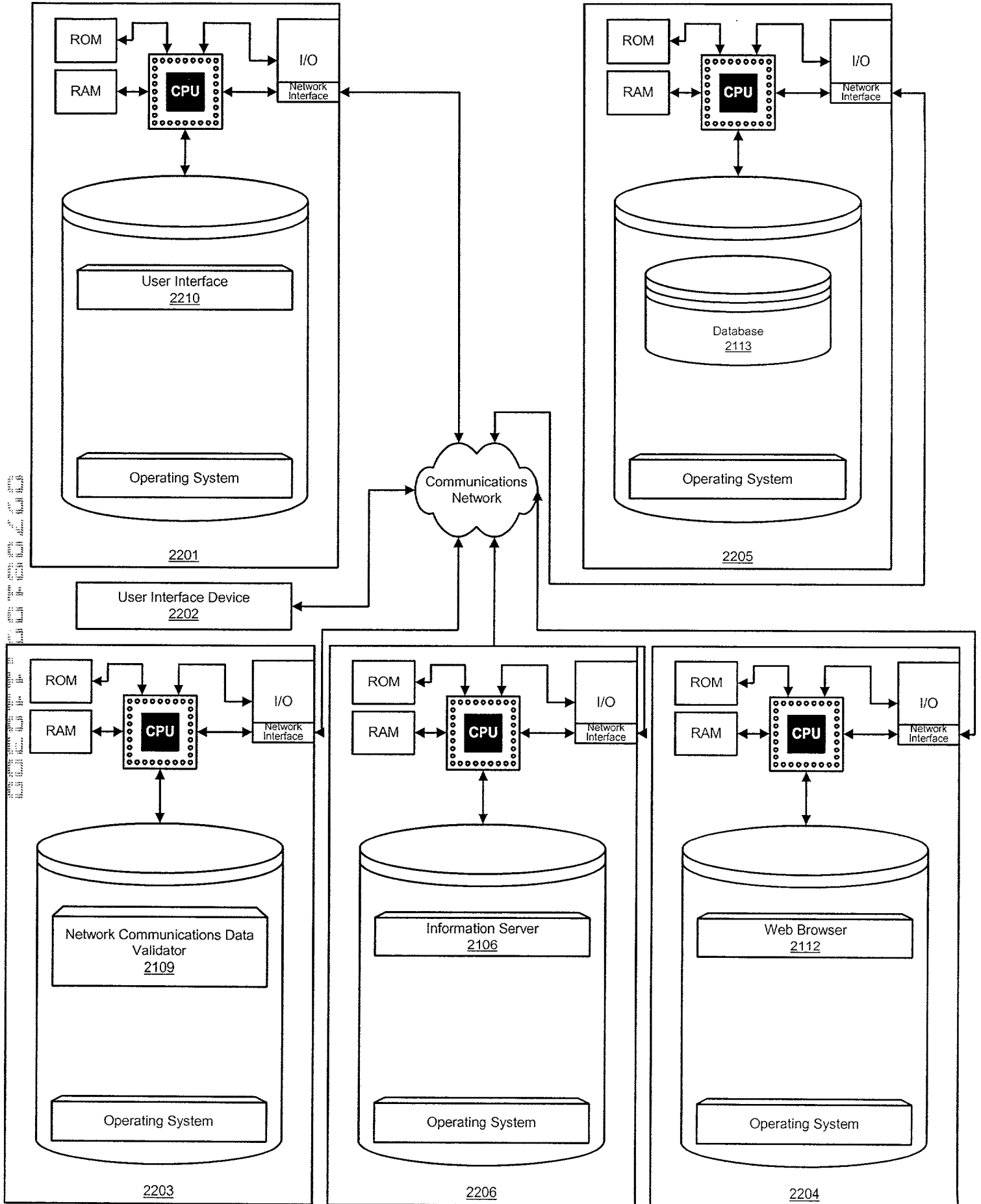


Figure 3

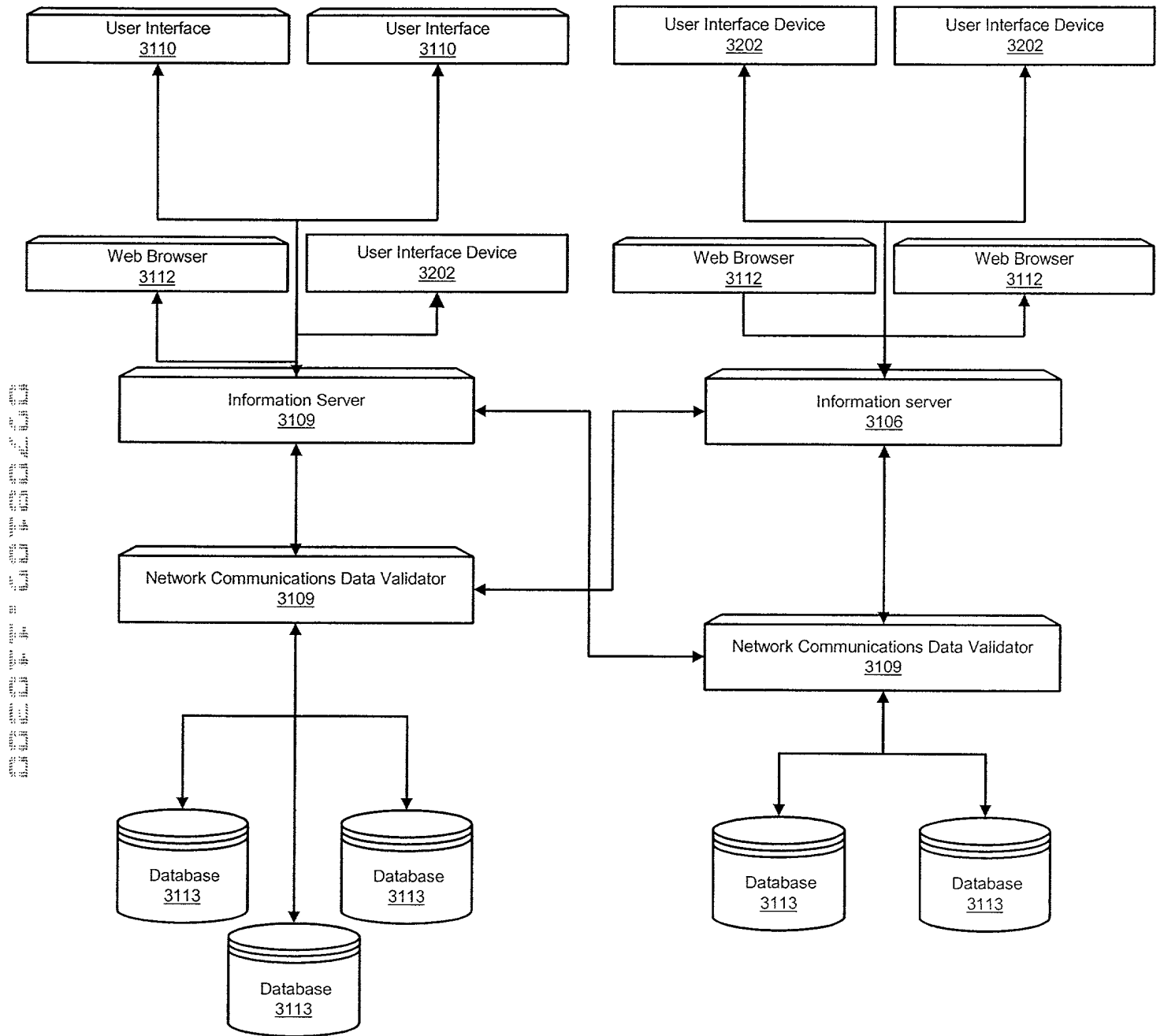


Figure 4

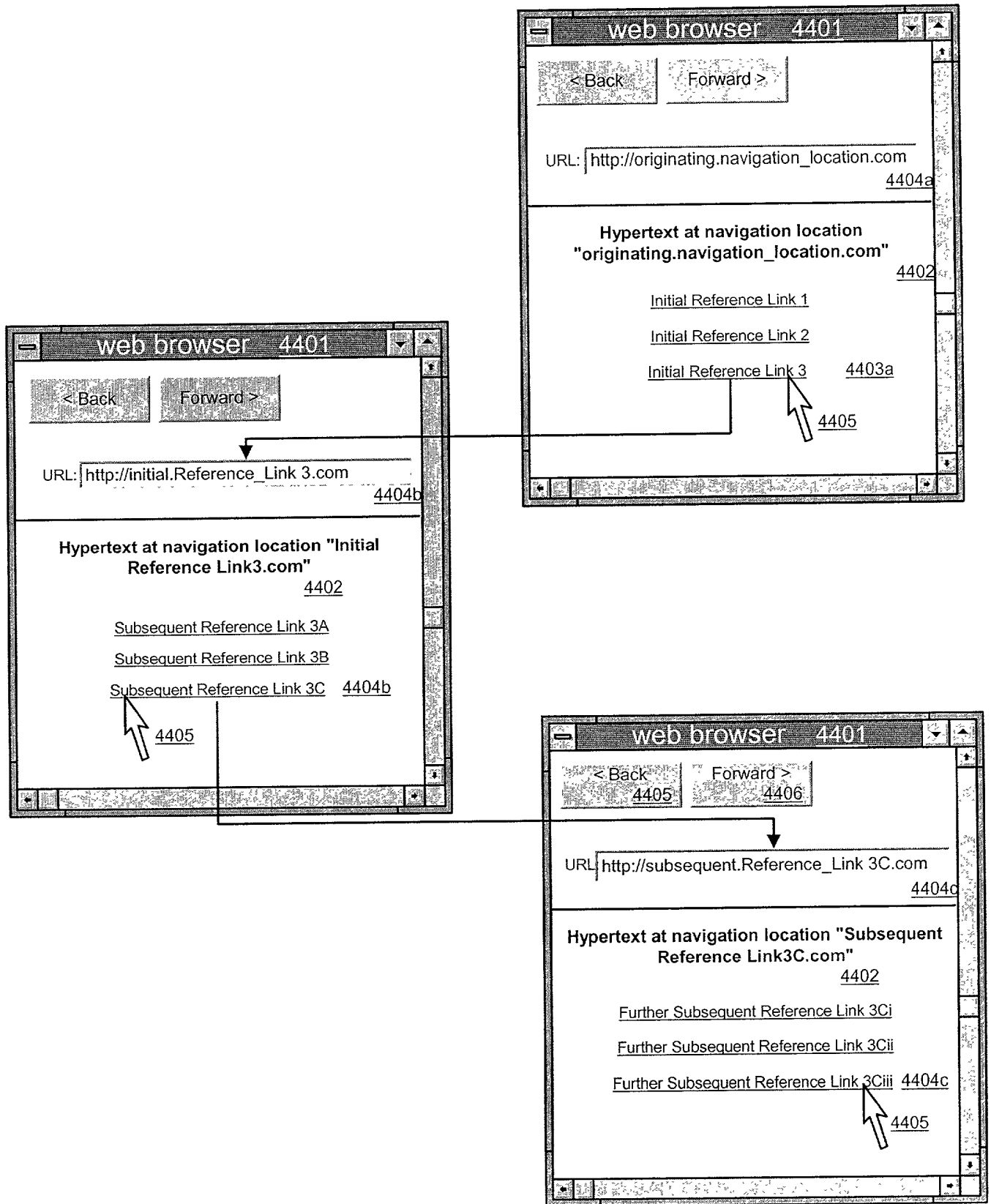


Figure 5

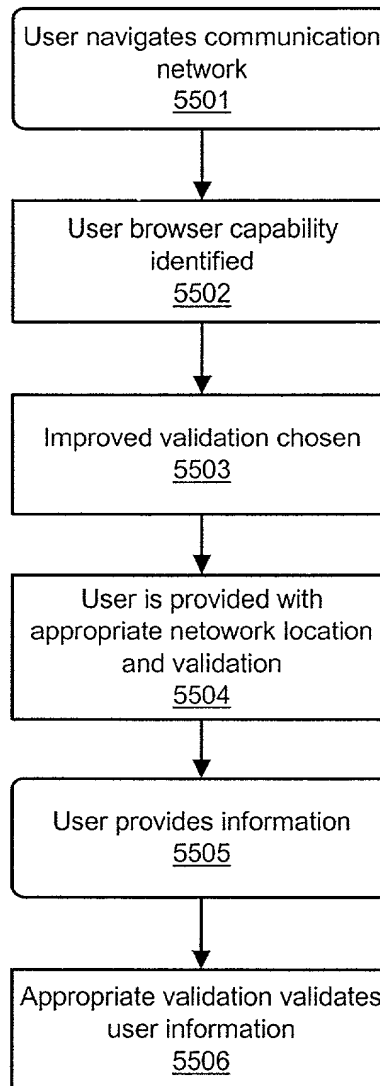
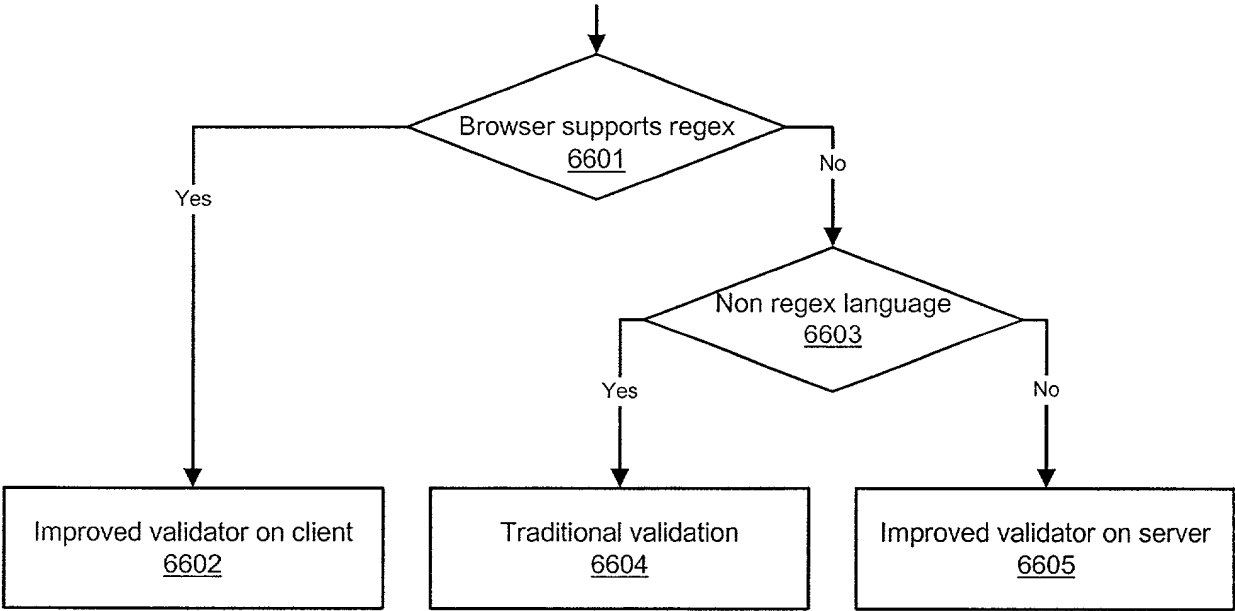


Figure 6



Improved Client Validator

Figure 7

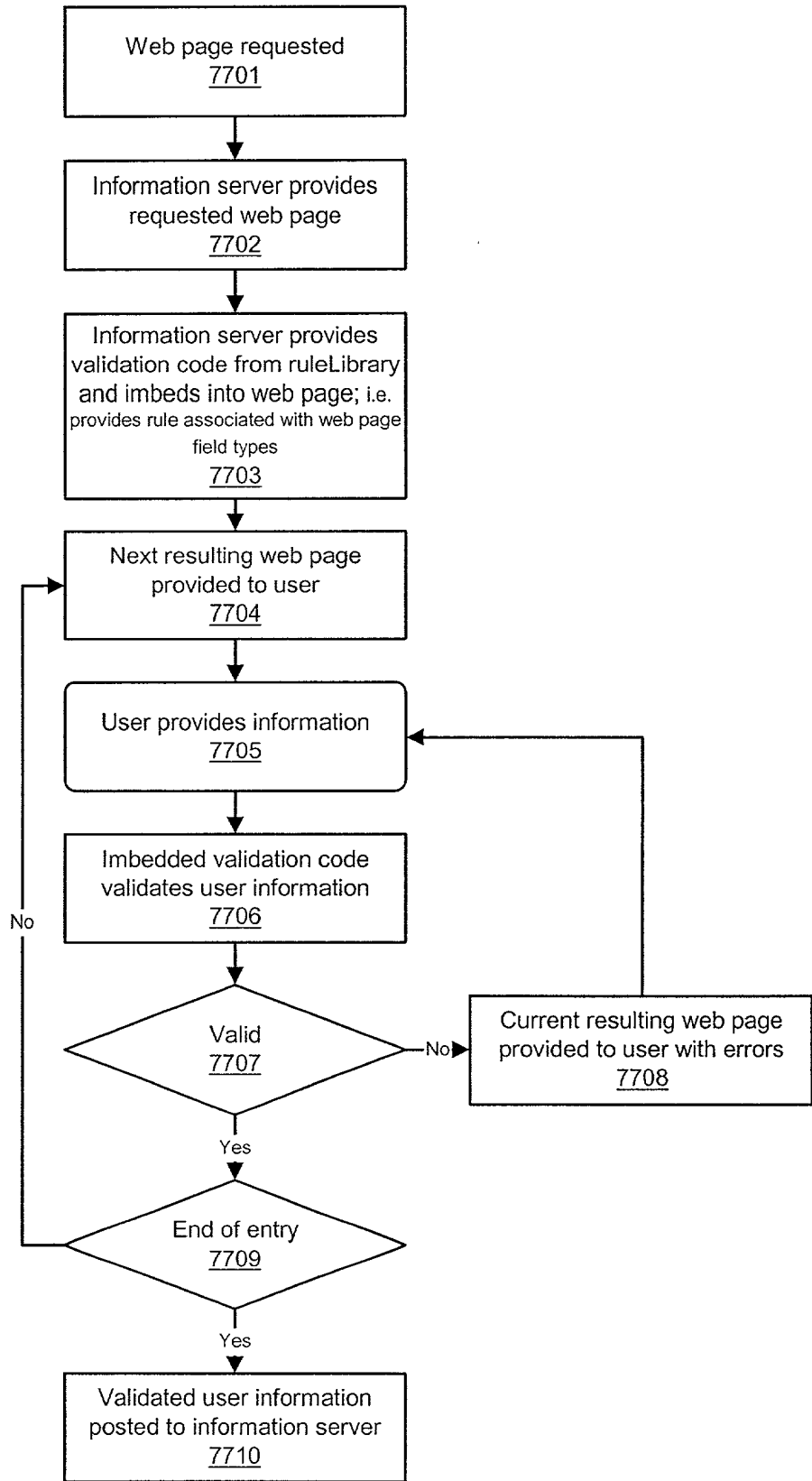


Figure 8

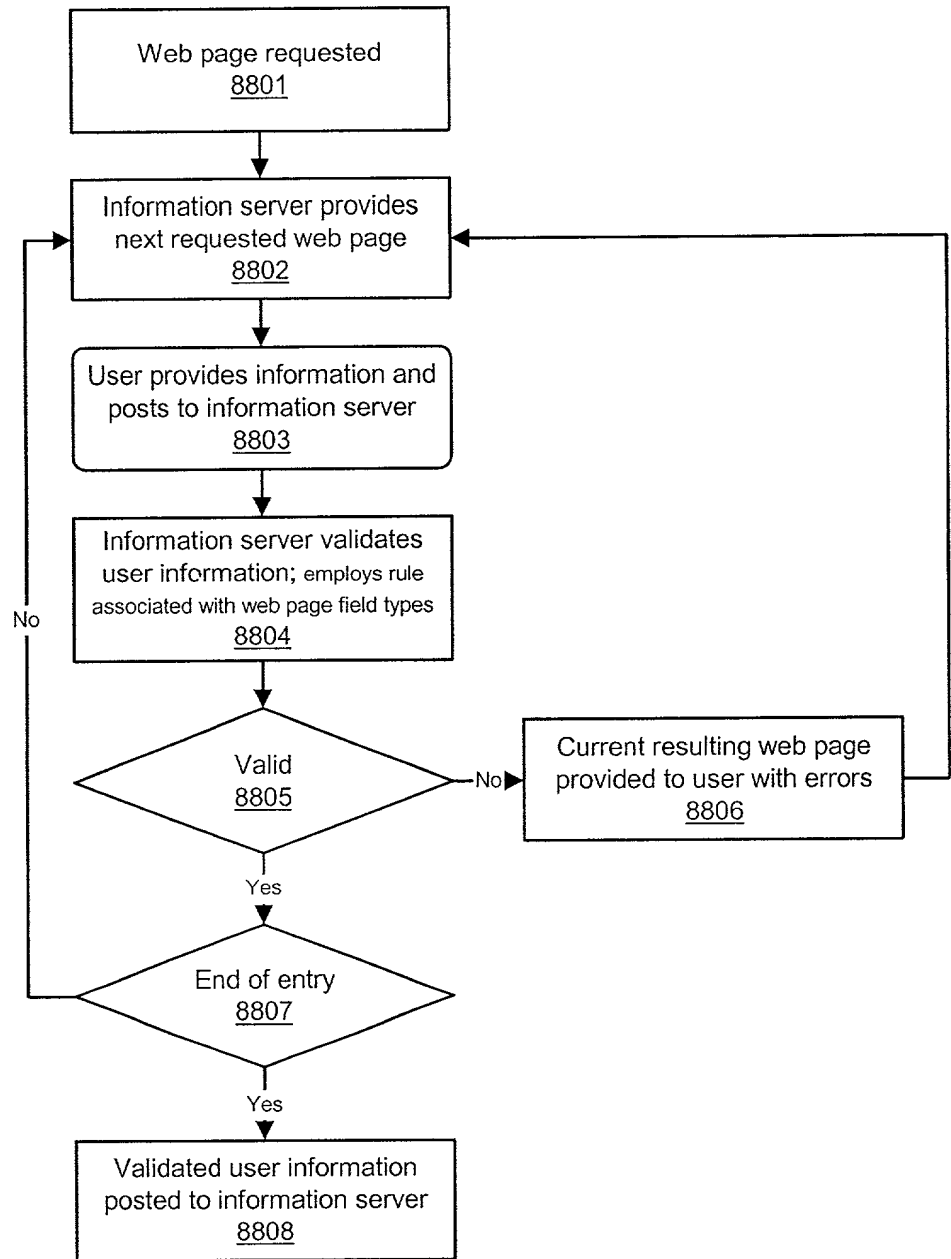
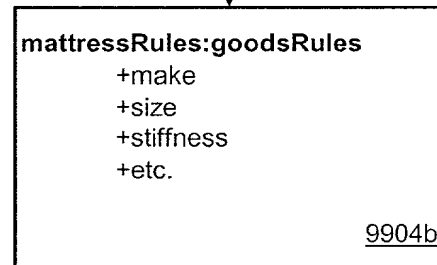
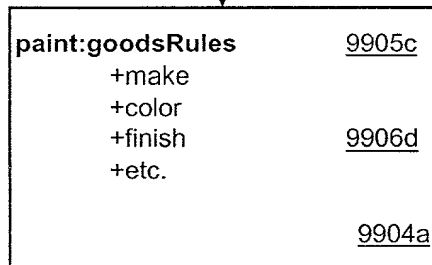
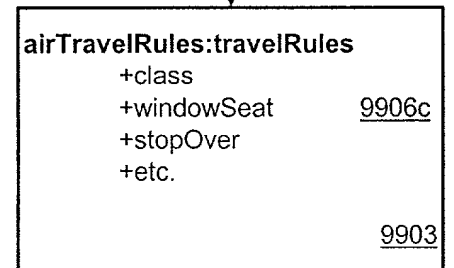
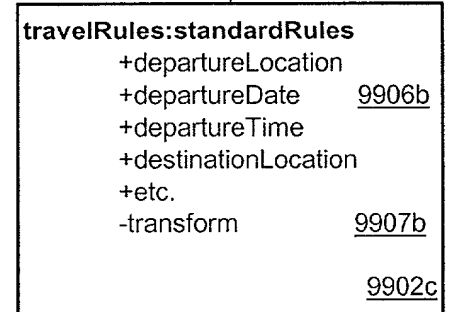
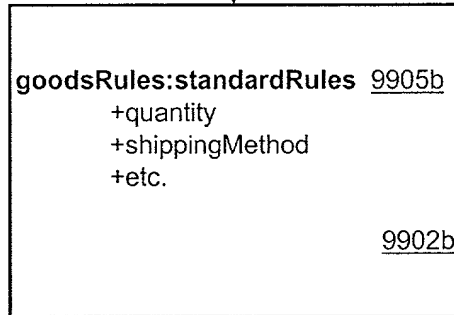
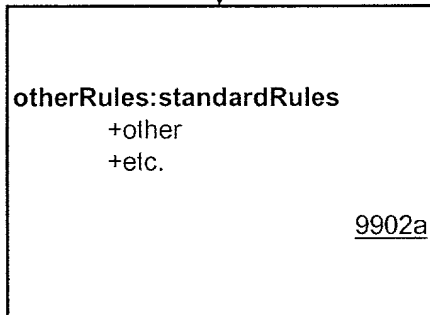
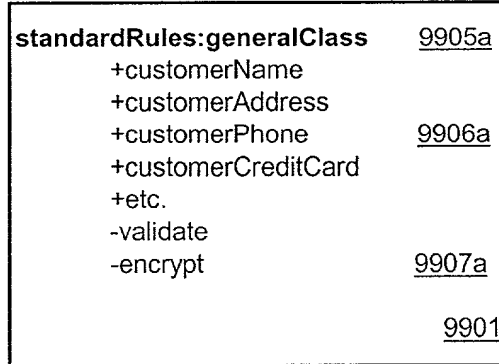


Figure 9

rulesLibrary



COMBINED DECLARATION AND POWER OF ATTORNEY FOR
ORIGINAL, DESIGN, NATIONAL STAGE OF PCT, SUPPLEMENTAL
DIVISIONAL, CONTINUATION OR CONTINUATION-IN-PART APPLICATION

As a below name inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

APPARATUS, SYSTEM, AND METHOD FOR VALIDATING NETWORK COMMUNICATIONS DATA
the specification of which

- a. ☒ is attached hereto
- b. ☐ was filed on _____ as application Serial No. _____ and was amended on _____ (if applicable).

PCT FILED APPLICATION ENTERING NATIONAL STATE

- c. ☐ was described and claimed in International Application No. _____ filed on _____ and as amended on _____ (if any).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, § 1.56(a).

I hereby specify the following as the correspondence address to which all communications about this application are to be directed:

SEND CORRESPONDENCE TO: John C. Andres, Esq.
priceline.com Incorporated
800 Connecticut Avenue
Norwalk, Connecticut 06854

DIRECT TELEPHONE CALLS TO: 203-299-8671

☐ I hereby claim foreign priority benefits under Title 35, United States Code § 119(a)-(d) or under § 365(b) of any foreign application(s) for patent or inventor's certificate or under § 365(a) of any PCT international application(s) designating at least one country other than the U.S. listed below and also have identified below such foreign application(s) for patent or inventor's certificate or such PCT international application(s) filed by me on the same subject matter having a filing date within twelve (12) months before that of the application on which priority is claimed:

☐ The attached 35 U.S.C. § 119 claim for priority for the application(s) listed below forms a part of this declaration.

<u>Country/PCT</u>	<u>Application Number</u>	<u>Date of filing (day, month, yr)</u>	<u>Date of Issue (day, month, yr)</u>	<u>Priority Claimed</u>
				[] YES [] NO
				[] YES [] NO
				[] YES [] NO

[] I hereby claim the benefit under 35 U.S.C. § 119(e) of any U.S. provisional application(s) listed below.

Provisional Application No.

Date of Filing (day, month, yr)

ADDITIONAL STATEMENTS FOR DIVISIONAL, CONTINUATION OR CONTINUATION-IN-PART
OR PCT INTERNATIONAL APPLICATION(S) (DESIGNATING THE U.S.)

I hereby claim the benefit under Title 35, United States Code § 120 of any United States application(s) or under § 365(c) of any PCT international application(s) designating the U.S. listed below.

<u>US/PCT Application Serial No.</u>	<u>Filing Date</u>	<u>Status (patented, pending, abandoned)/ U.S. application no. assigned (For PCT)</u>
<u>US/PCT Application Serial No.</u>	<u>Filing Date</u>	<u>Status (patented, pending, abandoned)/ U.S. application no. assigned (For PCT)</u>
<u>US/PCT Application Serial No.</u>	<u>Filing Date</u>	<u>Status (patented, pending, abandoned)/ U.S. application no. assigned (For PCT)</u>
<u>US/PCT Application Serial No.</u>	<u>Filing Date</u>	<u>Status (patented, pending, abandoned)/ U.S. application no. assigned (For PCT)</u>

[x] In this continuation-in-part application, insofar as the subject matter of any of the claims of this application is not disclosed in the above listed prior United States or PCT international application(s) in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application(s) and the national or PCT international filing date of this application.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or Imprisonment, or both, under Section 1001 of

Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I hereby appoint the following attorneys and/or agents with full power of substitution and revocation, to prosecute this application, to receive the patent, and to transact all business in the Patent and Trademark Office connected therewith: John C. Andres (Reg. No. 30,931) of priceline.com Incorporated, 800 Connecticut Avenue, Norwalk, Connecticut 06854

and

John A. Diaz (Reg. No. 19,550), John C. Vassil (Reg. No. 19,098), Alfred P. Ewert (Reg. No. 19,887), David H. Pfeffer, P.C. (Reg. No. 19,825), Harry C. Marcus (Reg. No. 22,390), Robert E. Paulson (Reg. No. 21,046), Stephen R. Smith (Reg. No. 22,615), Kurt E. Richter (Reg. No. 24,052), J. Robert Dailey (Reg. No. 27,434), Eugene Moroz (Reg. No. 25,237), John F. Sweeney (Reg. No. 27,471), Arnold I. Rady (Reg. No. 26,601), Christopher A. Hughes (Reg. No. 26,914), William S. Feiler (Reg. No. 26,728), Joseph A. Calvaruso (Reg. No. 28,287), James W. Gould (Reg. No. 28,859), Richard C. Komson (Reg. No. 27,913), Israel Blum (Reg. No. 26,710), Bartholomew Verdirame (Reg. No. 28,483), Maria C.H. Lin (Reg. No. 29,323), Joseph A. DeGirolamo (Reg. No. 28,595), Michael A. Nicodema (Reg. No. 33,199), Michael P. Dougherty (Reg. No. 32,730), Seth J. Atlas (Reg. No. 32,454), Andrew M. Riddles (Reg. No. 31,657), Bruce D. DeRenzi (Reg. No. 33,676), Michael M. Murray (Reg. No. 32,537) Mark J. Abate (Reg. No. 32,527) and Walter G. Hanchuk (Reg. No. 35,179) of Morgan & Finnegan, L.L.P. whose address is: 345 Park Avenue, New York, New York, 10154; and Edward A. Pennington (Reg. No. 32,588) of Morgan & Finnegan, L.L.P., whose address is 1775 Eye Street, Suite 400, Washington, D.C. 20006.

[] I hereby authorize the U.S. attorneys and/or agents named hereinabove to accept and follow instructions from _____ as to any action to be taken in the U.S. Patent and Trademark Office regarding this application without direct communication between the U.S. attorneys and/or agents and me. In the event of a change in the person(s) from whom instructions may be taken I will so notify the U.S. attorneys and/or agents hereinabove.

Full name of first joint inventor, if any Scott Nedderman

Inventor's signature* Scott Nedderman

Date

Residence 403 Elm Street, Monroe, Connecticut 06468

Citizenship USA

Post Office Address Same as residence

Full name of second inventor _____

Inventor's signature* _____ Date _____

Residence _____

Citizenship _____

Post Office Address _____

Full name of third inventor _____

Inventor's signature* _____ Date _____

Residence _____

Citizenship _____

Post Office Address _____

Full name of fourth inventor _____

Inventor's signature* _____ Date _____

Residence _____

Citizenship _____

Post Office Address _____

[] ATTACHED IS ADDED PAGE TO COMBINED DECLARATION AND POWER OF ATTORNEY FOR SIGNATURE BY THIRD AND SUBSEQUENT INVENTORS FORM.

* Before signing this declaration, each person signing must:

1. Review the declaration and verify the correctness of all information therein; and
2. Review the specification and the claims, including any amendments made to the claims.

After the declaration is signed, the specification and claims are not to be altered.

To the inventor(s):

The following are cited in or pertinent to the declaration attached to the accompanying application:

Title 37, Code of Federal Regulation, § 1.56

Duty to disclose information material to patentability.

(a) A patent by its very nature is affect with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is canceled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is canceled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

- (1) prior art cited in search reports of a foreign patent office in a counterpart application, and
- (2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

Title 35, U.S. Code § 101

Inventions patentable

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Title 35 U.S. Code § 102

Conditions for patentability; novelty and loss of right to patent

A person shall be entitled to a patent unless –

- (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for patent,

(b) the invention was patented or described in a printed publication in this or foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States, or

(c) he has abandoned the invention, or

(d) the invention was first patented or caused to be patented, or was the subject of an inventor's certificate, by the applicant or his legal representatives or assigns in a foreign country prior to the date of the application for patent in this country on an application for patent or inventor's certificate filed more than twelve months before the filing of the application in the United States, or

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent, or

(f) he did not himself invent the subject matter sought to be patented, or

(g) before the applicant's invention thereof the invention was made in this country by another had not abandoned, suppressed, or concealed it. In determining priority of invention there shall be considered not only the respective dates of conception and reduction to practice of the invention, but also the reasonable diligence of one who was first to conceive and last to reduce to practice, from a time prior to conception by the other ...

Title 35, U.S. Code § 103

Conditions for patentability; non-obvious subject matter

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Subject matter developed by another person, which qualifies as prior art only under subsection (f) or (g) of section 102 of this title, shall not preclude patentability under this section where the subject matter and the claimed invention were, at the time the invention was made, owned by the same person or subject to an obligation of assignment to the same person.

Title 35, U.S. Code § 112 (in part)

Specification

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise and exact terms also enable any person skilled in the art to which it pertains, or with which it is mostly nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Title 35, U.S. Code § 119

Benefit of earlier filing date in foreign country; right of priority

An application for patent for an invention filed in this country by any person who has, or whose legal representatives or assigns have, previously regularly filed an application for a patent for the same invention in a foreign country which affords similar privileges in the case of applications filed in the United States or to citizens of the United States, shall have the same effect as the same application would have if filed in this country on the date on which the application for patent for the same invention was first filed in such foreign country, if the application in this country is filed within twelve months from the earliest date on which such foreign application was filed; but no patent shall be granted on any application for patent for an invention which had been patented or described in a printed publication in any country more than one year before the date of the actual filing of the application in this country, or which had been in public use or on sale in this country more than one year prior to such filing.

Title 35, U.S. Code § 120

Benefit or earlier filing date in the United States

An application for patent for an invention disclosed in the manner provided by the first paragraph of section 112 of this title in an application previously filed in the United States, or as provided by section 363 of this title, which is filed by an inventor or inventors named in the previously filed application shall have the same effect, as to such invention, as though filed on the date of the prior application, if filed before the patenting or abandonment of or termination of proceedings on the first application or an application similarly entitled to the benefit of the filing date of the first application and if it contains or is amended to contain a specific reference to the earlier filed application.

Please read carefully before signing the Declaration attached to the accompanying Application.

If you have any questions, please contact John C. Andres.